Instructions for Including the $$ at the end of Riverpro Products

NWSI 1701/1702 require the inclusion of the double dollar sign ($$) at the end of the product content block of all products–segmented or not. The implementation date for this element is February 12, 2003. Products created in Riverpro can meet this requirement by simply modifying the appropriate templates. We describe possible options to meet this requirement below:

1. The Hydromet basis section is perhaps the least used section in Riverpro products. For those products that do not use the Basis section, one can create a new basis template as follows:

name: dollar_signs
phrasestr:$$

The Basis section does not require conditional statements. You can then re-define the product to include the Basis section as the last section in the product.

2. For products where the Call-to-Action section is the last section of the product, one can create a new CTA template:

name: dollar_signs
phrasestr: || $$

The cta template does not require conditional statements. The $$ is required to be on a line by itself. Since Riverpro concatenates all selected cta templates, the double pipe (||) is used to produce a new line . You can re-define the product to include the dollar_signs cta template as the last template in the cta section.

3. For products where the Roundup section is the last section, and the Basis section is already being used, one can add an additional line at the end of the appropriate template:

condition: TRUE
phrasestr: ||$$

The roundup requires a conditional statement prior to all phrase strings; hence, the condition: TRUE line. The double pipe (||) is used to produce a new line. Since this is an edit to an existing template, there is no need to re-define the product.

4. For products where the Impact Statement or Historical Crest Comparison in the last section, and the Basis section is already being used, one can add an additional line at the end of the appropriate template:

phrasestr: ||$$

These sections do not require conditional statements. The double pipe (||) is used to produce a new line. Since this is an edit to an existing template, there is no need to re-define the product.

5. For products where the tabular section is the last section in the product, and the points included in the product will vary from issuance to issuance (e.g an FLW or FLS), and the basis section is already being used, the following is some background and then guidance on how to modify the tabular template to include the $$:

In the tabular template, there are four keywords that tell the program to print information. These keywords are:

1. literal - which tells the application to print out the info immediately following the word literal

2. miscwrt - which tells the application to print out the information in the formats/varlist lines immediately preceding the miscwrt line

3. fp_id - which tells the application to print out the information in the formats/varlist lines immediately preceding the fp_id line for the forecast point location identified in the fp_id line

4. locid: which tells the application to print out the information in the formats/varlist lines immediately preceding the locid line for the location identified in the locid line. Not all variables work with the locid keyword.

When Riverpro was intially developed and tested at OUN prior to fielding in AWIPS, it only created products for forecast points. This was the case also in the early AWIPS builds. Also during those early days, the user had to have both fp_id lines in the template AND highlight the point in the Riverpro GUI in order for a location to be formatted in the product. Based on feedback from OUN, we removed the requirement for the fp_id line to be in the template. However, if you look through the message logs, you'll see warnings indicating a missing fp_id keyword when forecast points are included in the product by virtue of them being highlighted in the gui and the fp_id keyword is not in the template.

When Riverpro processes a tabular template, it reads down through the template, hitting all the keywords and outputting the data from top to bottom. When it's done with that process, it then goes back and sees which points are highlighted in the gui that haven't been processed, and uses the last formats/varlist pair in the template to process all remaining forecast points. It processes all the keywords first. If you do not have any fp_id keywords in the template, and have added a

literal:$$

at the end of the template, the application is processing the literal:$$, and then "doubling back" to pick up the forecast points highlighted in the gui.

To change this behavior, you'll need to add the fp_id keyword after the specific formats/varlist pair that outputs the point-specific data. Add an fp_id line for every forecast point. The application will only output data for those forecast points that are highlighted in the gui. However, inclusion of the fp_id line will force the write of the information for that point prior to the $$. Then add a literal:$$ at the end.